THE QL REPORT

PUBLISHED BY CURRY COMPUTER

VOL. 3, NO. 7

JULY 15, 1987

P.O. BOX 5607, GLENDALE, AZ 85312-5607

$15.00 FOR 12 ISSUES

Well, here we are in another summer. This is traditionally the time when it slows down for computer-related businesses. We remember past summers when the doldrums set in and there just wasn't a lot to talk about.

With the QL, however, there is a lot happening, both old and new. We are using SPELLBOUND now for all our documents and are absolutely amazed at the program. One thing we forgot to mention last issue was the fact that this is really a spelling corrector as well as a spelling checker. Suppose, for example, I begin to spell the word, "suppose" and can't remember whether it has one or two "p's". SPELLBOUND will beep if I only put in one. I then can press the control, shift, and "e" keys together and a list of words appears on the right hand side of the screen, alphabetically beginning with all words that start with "sup" . I press the "f5" key to continue seeing each word. When I see the word "suppose" on the screen and hopefully remember that is the correct way it is spelled, I press the enter key and "suppose" is entered right into the document. This is definitely a nice feature. The only criticism we have of the program is that for some reason when you turn on the capslock key, while you have SPELLBOUND activated, it slows down almost to a crawl. Except for that, it is a superb program and one that could easily cost twice as much as its $51.95 retail if it were on an IBM or a Macintosh.

Schoen is coming out with a new, IBM-AT style keyboard. The retail is going to be high, about $150.00. For those of you who are doing very serious word-processing you might want to consider this. Even with the high price of a keyboard such as this, using your QL in conjunction with SPELLBOUND would still make for an inexpensive system that would run circles around anything else on the market.

As more and more of you purchase the CP/Mulator, we would appreciate it if you could drop us a line when you find a new format. We are trying to keep on file all the various different ones to make it easier for people. It will save a lot of time if they know exactly what skew sectors, reserved tracks, etc. are needed for Kaypro, TRS-80, etc. And, if you have any public domain software that you have converted to QDOS format, let us know. As we said in last month's issue, some disk interfaces do not analyse CP/M disks correctly. If you already have the QDOS version then it doesn't matter- you will be able to run it in CP/M mode without difficulty.

We talked with Freddy Vacha recently of Digital Precision. They are working on a new screen dump program that will be compatible with EYE-Q. They are also working on a program that will be very similar to Talent's PCB series for circuit board design. As you can see, his DESKTOP PUBLISHER is a very nice product.

DESKTOP PUBLISHER is, quite simply, an amazing program. It took us about a day to master the basics. It uses what is called a "cameo" when you first boot up the program. The cameo shows one complete A4 size page. Although this size page (it is the size the QL User Guide uses) is quite common in England and Europe, it is obviously not as common over here as the 8 1/2 by eleven and Freddy has stated that future versions will hopefully over the option of this size. The first menu gives you options to edit the current page, list files, load and save pages, print, edit fonts, edit the QL sets, and switch tasks. We found it was very easy to exit DESKTOP, go into QUILL, for example, and then after quitting QUILL,get back into DESKTOP.

If you select as an option to edit the current page, the cameo shows a small window, roughly one-sixth (13%) the size of the whole paper, which you can move to any desired position in the cameo. The window is 400 by 256 pixels . There are "X" and "Y" co-ordinates to let you position the window exactly. Once this is done, the window you have chosen becomes the majority of the screen on your monitor.

On the right of the screen the current QL character set is displayed. Pressing "F5" will usually get you a help screen in case you forget some of the commands.

From the window you have selected on the cameo, you can then create any size window you desire within it for text entry. Before you can begin typing, you are asked for the character size you desire. DESKTOP does not word wrap in this mode which is unfortunate. However, after you have finished typing in the text in the specified window, you are asked if you would like the text justified. This option at least makes for a neater looking page. You also can change the character set (there are 8 to choose from) and change the stipples. If you are dissatisfied with you efforts, you can discard the text rather than keeping it and start from scratch.

Besides a text mode, there is also a "font" mode and this is very impressive. In DESKTOP a font is a character of much higher definition than any of the QL character sets. It consists of a 16 by 16 grid of graphics. There are 12 default fonts. At the beginning of the QL Report you will see some of the various types. On startup two of the fonts are dedicated to graphics patterns and ten are dedicated to letters and punctuation. These are not permanently fixed-

you can redesign them at will with the built in FONT DESIGNER.

Unlike the text mode where you have to create a window and can only insert text within this pre-defined area, while in font mode you can insert characters across the whole area of the cameo. You can change fonts at will, create windows for straight ql character set text, or go into graphics mode to create pictures.

At any time you can go back to the main menu and when at the top you will see the cameo displaying the entire A4 page. In most cases you will only see the general layout of the page- the cameo does not offer enough detail to read everything but at least with this method you can "see" what the total effect will be. We think it is a good idea.

The only serious problem we have had is importing QUILL files into DESKTOP. Trying to bring in straight QUILL documents also brings in all the machine code control codes. We have also found that saving the document to a file with a "lis" extension causes double line spacing to occur when imported into DESKTOP.

What we have had to do for this issue is set our page breaks to twenty-five lines which is the size of one full window in DESKTOP. Then, we load the "lis" file which we have edited through QED in QCODE. A "_lis" file from QUILL gives an extra carriage return which we cannot get rid of. If one uses THE EDITOR, the newest version contains a driver that will allow you to set parameters so that text goes into DESKTOP quite easily, supposedly, but we have not had a chance to check on this. FRONT PAGE EXTRA, again supposedly, will bring in QUILL documents and put them into columns automatically for you. However, when we tried this, it worked, but we still had a few problems- for one thing, we were missing characters from lines in the second column. This could be due to setting the right-hand margin to "75".

DESKTOP PUBLISHER gives a very professional-looking product. Digital Precision consistently is turning out high quality material for the QL. We also are impressed that they are the only major QL software producer that does not put copy protection on their software. All of it is very easily converted to disk, ram, etc. Next month we hope to be using THE EDITOR along with DESKTOP, to write QLR. If it is as easy to use as Freddy claims, it should be a very powerful combination.

We will be reviewing FRONT PAGE EXTRA in an upcoming issue. It is unfortunate that both desktop programs have deficiencies . We like the way FRONT PAGE EXTRA can bring in text into columns but as we said, earlier, it did not work perfectly for us. We like the way DESKTOP PUBLISHER makes use of a cameo, and the grid reference system and are very impressed by the fonts but give it low marks for it being unable to easily import QUILL files.

★★★★HELPFUL HINTS★★★★

We received the following letter and although have mentioned this tip in a previous issue of QLR, feel it should be stated again because it is useful. Mr. R.F Fraser-Smith writes:

"If, while in command mode or while a programme is running, you press CTRL & F5 together, it will stop the machine in its tracks; any key will restart it again. This could be extremely useful, as it works with both DIR and LIST, so that you can list the directory of a cartridge with a lot of files on it or a long program to the screen, and stop the scrolling at will to study a part of the directory or programme.

I can tell you, that I wish that I had known this a w or so ago, when I was just new to the QL. (see QLR March, 1986 .ed)

On another point, I called Motorola in Schaumburg, IL. last week, to ask them how I could get the instruction set for the 68008; I received today a set of literature which I haven't had time to study yet. I thought that I'd mention it, as an example of pleasantness and efficiency on their part. They didn't charge me anything at all!"

★★★★MICRODRIVE IMAGING★★

The TRUMP CARD and certain other boards have a very neat little method of loading files from microdrive cartridge called "Microdrive Imaging". It produces a microdrive image in your ramdisk. To do this, a ramdisk is formatted with the name of the microdrive required, for example: FORMAT ram1_mdv1 . To get an understanding of how useful this is, we ran a simple test. We turned on TOOLKIT II in TRUMP and used the "ucopy" command to copy the contents of QLSS from microdrive one to ram1. This took over two and one-half minutes. We then reset the machine and issued the command: FORMAT "ram1_mdv1". The microdrive kicked in for only ten seconds and then stopped. When we checked a directory of ram1_, all files were there!

The speed of microdrive imaging is very very fast. The ramdisk can even load a microdrive with a damaged directory. It cannot, however, load a microdrive with a damaged s The ramdisk will try up to three times to read a fa sector. If it fails, the number of good sectors returned from the format will be fewer than the total number. Any file with bad sectors will be marked with an "###" in the ramdisk directory.

For those of you contemplating buying the TRUMP CARD even though you do not at the moment have diskdrives, this feature alone is indispensable when working only with microdrives, and ramdisk.

In somewhat related ne us, Amstrad has been in the newspapers recently. The WSJ reported in late June that the company's price per share of stock had plummeted 23% in recent weeks. Chase Manhatten Securities are said to have revised their 1987/88 forecasts for Amstrad downwards. Chase cut its profits forecast from £214 million to £180 million and others cut it from £175m to £150m . Their PC 1512, an IBM clone, is not selling as well as had been expected. The sales are only half of last September's 70,000 target. Amstrad is set to release their new PC1640 in late August and rumours have been circulating that they will drop the the 1512 altogether.

Perhaps Amstrad will at least consider the possibility of putting the QL back into production. The Spectrum still has, at last look, about a 43% market share in the U.K., almost double its closest competitor , Commodore.

ULTRASOFT GCK

by Rich Bazan

There are many toolkits in the market for the QL. All of them offer some amount of extensions dedicated to graphics. The Ultrasoft Graphic Construction Toolkit (GCK) is a newcomer that is entirely dedicated to graphics and offers several unique features compared to the others.

The code for the toolkit is about 12k in length making it suitable for use on unexpanded machines.Within this space are contained over 60 graphic extensions to Superbasic.In their ads Ultrasoft indicate that the toolkit can be compiled to produce a graphics program with the power of Digital Precision's Eye-Q in about 8k of code.Indeed such a program is included with the package in the form of a basic program which can be modified,if desired,and then compiled using either Turbo or Liberator.

The key distinction I found in the GCK is that many of the extensions provided can be supplied with normal calling parameters to define such things as screen locations,block size,colors,etc. If parameters are not included many of the extensions provide for a powerful interactive mode to be invoked. For instance once a screen section is defined, using the command ROLL will cause the selected window's contents to roll to the left,right,up,or down using thru use of the appropriate arrow keys.Similiar functions are provided for mirroring the contents vertically or horizontally as well as in the defintion of the windows themselves.This makes for easy graphics manipulation.

The program will allow for the storage,retrieval and manipulation (memory permitting!) of 15 normal basic screens as well as 15 screen banks.These are off-screen storage areas which can be manipulated without being visible.Also provided is a compression/decompression capability allowing for the definition of 15 compressed screens and 15 compressed banks.This provides a tremendous medium for creative windowing.

Manipulations are also available for magnifying or shrinking screens and banks. Included in this are extensions for handling Spectrum screens once they have been transferred to the QL.

More routine extensions are provided for plotting points,for reading the color of selected pixels and for reading items such as screen mode ,screen or bank status and cursor positioning.

An extension is provided to reclaim fragmented space in the common heap(garbage collection).Also provided is one which will cause a user-defined message to be scrolled across the screen in ticker-tape style while awaiting a keyboard input.

This Ultrasoft package makes manipulating screens and shapes easy and is strongly recommended for graphically-inclined QL users.

SPINNING RINGS

by Marshall Stiles

Every now and then it is a good idea to explore or review a command in detail. The listing of spin_ring_bas is included below in order to demonstrate some of the aspects of the SuperBASIC command, CIRCLE.

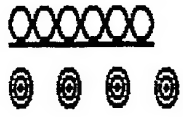
Not counting the channel indicator, the CIRCLE command has five parameters. For this article the default channel (#1) is used, therefore, no channel is specified. The first three parameters must be supplied. The last two will default to the values needed to produce a circle, as opposed to an ellipse.

The first parameter is the position of the center of the circle in the window it is to be drawn in, counting from left to right (or on the x-axis). The second is the position from the bottom of the window counting towards the top of the window (or on the y-axis). The third is the radius or distance from the center of the circle to the outer edge. The fourth is called eccentricity. This parameter is what causes elliptical or egg shapes. The ellipse is formed in conjunction with the length of the radius, so that the elongated portion (the furthest point from center on the rim of the ellipse) is equal to the value specified for the radius (third) parameter. The fifth parameter, angle, determines the direction or angle of the axis.

In line 140 of spin_ring_bas "ecc" represents eccentricity, and the
FOR loop spins the rings on the axis ("angle"). Note that eccentricity is a number between -1 and 1. This short span "spins" the rings in a full 360 degree spin. Actually, this is just an illusion. The rings are just being redrawn as differing ellipses. The angle parameter alternates between PI and PI*.5 (or PI/2) causing perpendicular spins. All possible axis angles (0 - 360 degrees) can be represented with values between 0 and PI or values from 0 to 360.

Pressing a key while spin_ring_bas is running will eventually halt the program. You might want to do some experimentation. For example, try changing "angle" (line 110) to another value. At any rate, enjoy the program!

```
100   REMark  spin_ring_bas,  Marshall  Stiles,  1987
110     PAPER     4:CLS:PAPER     0:X=50:Y=50:radius=25:
angle=PI*.5: ecc=0
120 REPeat outer_loop
130 REPeat loop
140 FOR ecc=-1 TO 1 STEP 5E-2
150 INK 7:CIRCLE x,y,radius,ecc,angle
160 INK 0:CIRCLE x,y,radius,ecc1,angle
170 ecc1=ecc
180 IF ecc>=1
190 IF angle=PI*.5
200     angle=PI
210     EXIT loop
220   END IF
230   IF angle=PI
240     angle=PI*.5
250     EXIT loop
260   END IF
270  END IF
280 END FOR ecc
290 END REPeat loop
300 IF INKEY$<>"" THEN INK 7:EXIT outer_loop
310 END REPeat outer_loop
```



ADDRESSING DEVICES IN PASCAL ★★★

by Mel Mac Karon

Pascal is a language that was designed to work with files, rather than channels (as BASIC does). In general terms, this should make no difference in most cases, but it does when we want to send output to a different location.

With SuperBASIC, if we want to send output to the printer, we have to direct the bit stream to a specific channel which is tied to a string identifier. Therefore, to go to the printer, we could use the following lines:

```
10 OPEN #3, ser1z
20    PRINT    #3,    'It's    a    beautiful    morning.'
30 CLOSE #3
```

This would print the above phrase to the device attached to ser1. If we wanted to insert lines within the program that did not go to the printer, we would just omit #3 after the PRINT command.

Mechanically, Pascal works in a similar manner, but we must do a bit more coding to get there. However, the increase in speed makes the work quite worthwhile . The following code will accomplish the same thing as we did above:

```
program          testprint(          input,output,printfile);

{This  program  sends  its  output  to  the  printer}


const
printer = 'ser1z';
type
iodev = packed array [1..5] of char;
{identify I/O dev}
var
printout : iodev;
printfile : text; {other program file}

begin {testprint}

printout     :=     printer;     {assign     I/O     device}

rewrite(printfile,printout);
{init. output file}

writeln(printfile,'It's a beautiful
morning.')

end. {testprint}
```

As the above listing shows, you must first declare a textfile in addition to the standard Pascal textfiles, input and output. This is declared as a var of type text. Then, once we get into the main body of the program, we assign a string constant(ser1z) to the file. This is accomplished by use of the rewrite command, which opens the output channel. Finally,the actual printing is accomplished in much the same way as we did in SuperBASIC.

To get a small idea of the speed with which Pascal works (and also demonstrate that you can control where you send your output), type in the following listing. Before you run it, of course, you must compile it and exec_w paslink to link it into the runtime library. You will also need a formatted microdrive cartridge in drive one:

```
program testinout(input,output,outdev);
 {This program tests sending output to different output
devices- either to the printer or to and from mdv1_.}

    const

       printer  = 'ser1z     ';  {5 spaces after
                                   'ser1z'}
       drive ='mdv1_short';        {name of file}
    type
       io = packed array [1..10] of char;  {identifies
                                            I/O device}
    var
       sendout : io;
       ch : char;
       outdev : text;
    begin {testinout}
       sendout := printer;      {assign I/O device}
        rewrite(outdev,sendout); {initialize output
                                  file}
       writeln(outdev,'It's a beautiful morning.');
        writeln(outdev,'I  enjoy  talking  to  my
              printer.');
       writeln(outdev);          {print blank line}
       writeln(outdev,'Goodbye for now, printer.');
        sendout  :=  drive;          {assign  I/O  to
                                      m-drive}
        rewrite(outdev,sendout);    {initialize  new
                                     file}

    writeln(outdev,'I am now speaking to drive one.');
       reset(outdev,sendout);     {set to beginning of
                                   file}
       while not  eof(outdev) do    {while not  end of
                                     file}
      begin
            read(outdev,ch);      {read one character
                                   at a time}
            write(ch)   {send  each char.to screen}
      end;

       writeln;
       writeln;
       writeln('(A successful test)')

    end. {testinout}
```

{Ed. note: all items between "()" are comments and not needed in the program}.

Note in the above listing that, when we took the message off the disk, we had to do it one character at a time. That is a reflection of working with packed arrays. We may write them out all at once, but only read them in bit by bit.

When you run this, you should find a tremendous increase in speed. You should see the message telling that the program is finished before your cartridge drive finishes its rundown.

Next month we will have more news on desktop publishing with the QL. We hope to compare graphics with the various desktop packages and see how they compare. We also will be reviewing some new utility programs.

Until next month, enjoy your QL.